



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/667,542	09/22/2003	Robert Anthony DeLine	MS304414.1 /MSFTP464US	9926
27195	7590	02/08/2007	EXAMINER	
AMIN, TUROCY & CALVIN, LLP 24TH FLOOR, NATIONAL CITY CENTER 1900 EAST NINTH STREET CLEVELAND, OH 44114			YIGDALL, MICHAEL J	
			ART UNIT	PAPER NUMBER
			2192	
SHORTENED STATUTORY PERIOD OF RESPONSE		MAIL DATE	DELIVERY MODE	
3 MONTHS		02/08/2007	PAPER	

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	10/667,542	DELIN ET AL.
	Examiner	Art Unit
	Michael J. Yigdall	2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 22 September 2003.  
 2a) This action is FINAL.                    2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-25 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 1-25 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on 22 September 2003 is/are: a) accepted or b) objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) Notice of References Cited (PTO-892)  
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)  
 3) Information Disclosure Statement(s) (PTO/SB/08)  
 Paper No(s)/Mail Date 4/23/04, 8/29/05.

4) Interview Summary (PTO-413)  
 Paper No(s)/Mail Date. \_\_\_\_\_.  
 5) Notice of Informal Patent Application  
 6) Other: \_\_\_\_\_.

## **DETAILED ACTION**

1. Claims 1-25 are pending. A priority date of September 22, 2003 is considered.

### ***Specification***

2. The use of the trademarks "JAVA" (see, e.g., page 6, line 10) and "VISUAL BASIC.NET" (see, e.g., page 8, line 4) has been noted in this application. All trademarks, including those noted here and any others used in the application, should be capitalized wherever they appear and should be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner that might adversely affect their validity as trademarks.

### ***Double Patenting***

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In*

*re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1, 15-17 and 19-25 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 12, 13, 15, 16 and 21 of copending Application No. 10/681,759.

Although the conflicting claims are not identical, they are not patentably distinct from each other because they recite analogous subject matter. For example, the following table compares claim 1 of the present application to claim 1 of the copending application:

Present Application

1. An executable code check system comprising:  
an input component that receives an object file having an embedded specification;

Copending Application

1. An executable code check system comprising:  
an input component that receives an object file and a specification associated with

and,

a checker that employs the specification to facilitate static checking of the object file, the checker providing information if a fault condition is determined.

the object file, the specification comprising information associated with a plug-in condition for a method; and,

a checker that employs the specification to facilitate static checking of the object file, the checker providing information if a fault condition is determined.

As illustrated, every feature recited in claim 1 of the present application is similarly recited in claim 1 of the copending application, except for the limitation that the specification is an “embedded specification.” Nonetheless, claim 12 of the copending application recites, “The system of claim 1, wherein the specification is embedded with the object file.” Thus, claim 12 of the copending application anticipates claim 1 of the present application.

The analysis of the other conflicting claims is analogous. For example, claim 15 of the present application is similar to claim 1 but recites a limitation that the specification is “stored in a specification repository.” Claim 13 of the copending application anticipates this claim.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

#### ***Claim Rejections - 35 USC § 101***

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 1-25 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

With respect to claims 1-14, the claims are directed to an “executable code check system.” However, as recited, the system amounts to software *per se*. The claims do not recite any computer hardware components that would permit the functionality the system to be realized. See MPEP § 2106.01(I). Furthermore, claimed subject matter lacks a practical application because, as recited, the claims do not produce a useful, concrete and tangible result. Claim 1 recites, “the checker providing information if a fault condition is determined.” The claimed subject matter does not produce (1) a useful result because it does not reflect at least one practical utility described in the specification, such as providing the information to a programmer in the form of an error message so that the programmer can correct the error (see, e.g., page 10, lines 35-40). Likewise, the claimed subject matter does not produce (2) a tangible result because the final result (i.e., the provided information) is abstract in nature and is not limited to a result with real-world value. The claimed subject matter does not produce (3) a concrete result because the information is provided only if a fault condition is determined. In other words, the final result is not assured and repeatable because, as recited, nothing is provided if a fault condition is not determined. See MPEP § 2106(IV). Claims 2-14 do not remedy claim 1 with respect to the issue of non-statutory subject matter.

With respect to claim 15 and 16, the claims are directed to an “executable code check system.” However, as recited, the system amounts to software *per se*. Furthermore, the claimed

subject matter lacks a practical application because, as recited, the claims do not produce a useful, concrete and tangible result. See the analysis of claims 1-14 above.

With respect to claims 17-19, the claims are directed to a “method of facilitating static checking of executable code.” However, the claimed subject matter lacks a practical application because, as recited, the claims do not produce a useful, concrete and tangible result. Claim 17 recites, “providing information associated with the fault condition, if a fault condition is determined to exist.” The claimed subject matter does not produce (1) a tangible result because the final result (i.e., the provided information) is abstract in nature and is not limited to a result with real-world value. The claimed subject matter does not produce (2) a concrete result because the information is provided only if a fault condition is determined to exist. In other words, the final result is not assured and repeatable because, as recited, nothing is provided if a fault condition is determined not to exist. See MPEP § 2106(IV). Claims 18 and 19 do not remedy claim 17 with respect to the issue of non-statutory subject matter.

With respect to claims 20 and 21, the claims are directed to a “method of facilitating static checking of executable code.” However, the claimed subject matter lacks a practical application because, as recited, the claims do not produce a useful, concrete and tangible result. See the analysis of claims 17-19 above.

With respect to claim 22, the claim is directed to a “data packet transmitted between two or more computer components that facilitates static checking of executable code.” However, as recited, the data packet amounts to software *per se*. The claim does not recite that the data packet is embodied in such a manner that would permit the functionality of the data packet to be

realized. See MPEP § 2106.01(I). Furthermore, claimed subject matter lacks a practical application because, as recited, the claim does not produce a useful, concrete and tangible result. Claim 22 recites, “the embedded specification providing information to be employed to statically check the executable code.” The claimed subject matter does not produce a tangible result because the final result (i.e., the provided information) is abstract in nature and is not limited to a result with real-world value. See MPEP § 2106(IV).

With respect to claim 23, the claim is directed to a “data packet transmitted between two or more computer components that facilitates static checking of executable code.” However, as recited, the data packet amounts to software *per se*. The claim does not recite that the data packet is embodied in such a manner that would permit the functionality of the data packet to be realized. See MPEP § 2106.01(I). Furthermore, claimed subject matter lacks a practical application because, as recited, the claim does not produce a useful, concrete and tangible result. Claim 23 recites, “a specification that provides information to be employed to statically check the executable code.” The claimed subject matter does not produce a tangible result because the final result (i.e., the provided information) is abstract in nature and is not limited to a result with real-world value. See MPEP § 2106(IV).

With respect to claim 24, the claim is directed to a “computer readable medium storing executable components of an executable code check system.” However, the claimed subject matter lacks a practical application because, as recited, the claim does not produce a useful, concrete and tangible result. Claim 24 recites, “the checker providing information if a fault condition is determined.” The claimed subject matter does not produce (1) a useful result

because it does not reflect at least one practical utility described in the specification, such as providing the information to a programmer in the form of an error message so that the programmer can correct the error (see, e.g., page 10, lines 35-40). Likewise, the claimed subject matter does not produce (2) a tangible result because the final result (i.e., the provided information) is abstract in nature and is not limited to a result with real-world value. The claimed subject matter does not produce (3) a concrete result because the information is provided only if a fault condition is determined. In other words, the final result is not assured and repeatable because, as recited, nothing is provided if a fault condition is not determined. See MPEP § 2106(IV).

With respect to claim 25, the claim is directed to an “executable code check system.” However, the claimed subject matter lacks a practical application because, as recited, the claim does not produce a useful, concrete and tangible result. Claim 25 recites, “means for providing information if a fault condition is determined to exist.” The claimed subject matter does not produce (1) a useful result because it does not reflect at least one practical utility described in the specification, such as providing the information to a programmer in the form of an error message so that the programmer can correct the error (see, e.g., page 10, lines 35-40). Likewise, the claimed subject matter does not produce (2) a tangible result because the final result (i.e., the provided information) is abstract in nature and is not limited to a result with real-world value. Finally, the claimed subject matter does not produce (3) a concrete result because the information is provided only if a fault condition is determined to exist. In other words, the final result is not assured and repeatable because, as recited, nothing is provided if a fault condition is determined not to exist. See MPEP § 2106(IV).

***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-8 and 11-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over “Enforcing High-Level Protocols in Low-Level Software” by DeLine et al. (art of record, “DeLine”) in view of U.S. Patent No. 5,854,924 to Rickel et al. (“Rickel”).

With respect to claim 1, DeLine discloses an executable code check system (see, for example, page 1, Abstract) comprising:

an input component that receives a file having an embedded specification (see, for example, page 1, section 1, “The Vault programming language ...” et seq., which shows receiving a file having an embedded specification); and,

a checker that employs the specification to facilitate static checking of the file, the checker providing information if a fault condition is determined (see, for example, page 1, Abstract, which shows employing the specification to facilitate static checking of the file, and page 1, section 1, “Vault’s type checker exhaustively seeks and reports any violation of such a protocol,” which shows providing information if a fault condition is determined).

DeLine further discloses that the system operates at compile time (see, for example, page 7, section 4, "... Vault's type checker catches at compile time many of the errors ..."), but does not expressly disclose that the file is an object file.

However, in an analogous art, Rickel discloses an executable code check system (see, for example, the abstract). Rickel further discloses receiving an object file, employing a specification to facilitate static checking of the object file, and providing information if a fault condition is determined (see, for example, column 3, lines 18-26). The system is independent of the original programming language (see, for example, column 4, lines 35-39).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of DeLine to operate on an object file, as Rickel suggests. For example, one of ordinary skill in the art would have been motivated to modify the system of DeLine such that it is independent of the original programming language.

With respect to claim 2, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses the checker further removing the embedded specification from the object file (see, for example, DeLine, page 2, section 2.1, "Since guards and keys are purely compile-time entities, the function foo will be compiled into a function taking an ordinary FILE parameter and an ordinary int parameter," which shows that the embedded specification is removed from the object file).

With respect to claim 3, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses the specification comprising information associated with a method that performs at least one of allocation and release of a resource (see, for example, DeLine, page 4,

section 2.2, “Figure 2 shows three functions that use this region abstraction …” et seq., which shows that the specification comprises information associated with a method that allocates and releases a resource).

With respect to claim 4, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses the specification comprising information associated with an order in which methods of an object can be called (see, for example, DeLine, page 1, section 1, “Such a protocol can specify that operations must be performed in a certain order …” et seq., which shows that the specification comprises information associated with an order in which methods can be called).

With respect to claim 5, the rejection of claim 4 is incorporated, and DeLine in view of Rickel further discloses that method order is constrained by specifying a finite state machine in which the states have symbolic names and transitions between states are labeled with method names (see, for example, DeLine, page 4, section 2.3, “This interface uses the ability for keys to have states to enforce the necessary steps …” et seq., which shows a finite state machine with states that have symbolic names such as “raw” and “named” and transitions that are labeled with method names such as “bind” and “listen”).

With respect to claim 6, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses the specification comprising a state-machine protocol wherein a method specifies a pre-state and a post-state (see, for example, DeLine, page 2, section 2.1, “In Vault, a function’s type has a pre- and post-condition …” et seq., which shows that a method specifies a pre- and post-state in the specification).

With respect to claim 7, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses the specification comprising information associated with a transition of a finite state machine (see, for example, DeLine, page 4, section 2.3, “This interface uses the ability for keys to have states to enforce the necessary steps …” et seq., which shows that the specification comprises information associated with a transition of a finite state machine).

With respect to claim 8, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses the specification comprising at least one of a rule using an interface, system resource management, order of method calls and formatting of a string parameter (see, for example, DeLine, page 1, section 1, “The Vault programming language …” et seq., which shows that the specification comprises system resource management).

With respect to claim 11, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses the specification comprising information associated with a state-machine protocol (see, for example, DeLine, page 4, section 2.3, “This interface uses the ability for keys to have states to enforce the necessary steps …” et seq., which shows that the specification comprises information associated with a state-machine protocol).

With respect to claim 12, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses the specification comprising an attribute associated with at least one of a field and a parameter providing information associated with whether or not the at least one of a field and a parameter can be aliased (see, for example, DeLine, page 6, section 3.1, “The key to ensuring that a program does not reference a resource after that resource has been released …” et

seq., which shows that the specification comprises an attribute associated with a field that provides information associated with whether the field can be aliased).

With respect to claim 13, the rejection of claim 1 is incorporated, and DeLine in view of Rickel further discloses that the specification facilitates modeling of a heap modeling (see, for example, DeLine, page 3, section 2.2, “A typical C program …” et seq., which shows that the specification facilitates heap modeling).

With respect to claim 14, the rejection of claim 13 is incorporated, and DeLine in view of Rickel further discloses the checker employing an algorithm that performs a data flow analysis over the heap model comprising a typing environment and a set of capabilities (see, for example, DeLine, pages 6-7, section 3.3, “Existential types are useful for encoding that certain values carry capabilities …” et seq., which shows performing an analysis over the heap model that comprises a typing environment and a set of capabilities).

With respect to claim 15, the claim is directed to an executable code check system, the elements of which are analogous to those of claim 1 (see the rejection of claim 1 above).

DeLine in view of Rickel further discloses the specification stored in a specification repository (see, for example, Rickel, column 4, lines 32-35, which shows that the specification is stored in a library or repository).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of DeLine such that it operates with a specification stored in a repository, as Rickel suggests. For example, one of ordinary skill in the art would have been motivated to provide the system of DeLine with the flexibility to retrieve the specification from a

repository external to the object file, so as employ the same specification to facilitate static checking of several object files.

With respect to claim 16, the rejection of claim 15 is incorporated, and DeLine in view of Rickel further discloses the system further comprising the specification repository (see, for example, Rickel, FIG. 1a).

With respect to claim 17, the claim is directed to a method of facilitating static checking of executable code, the elements of which are analogous to those of claim 1 (see the rejection of claim 1 above).

With respect to claim 18, the rejection of claim 17 is incorporated, and DeLine in view of Rickel further discloses removing the embedded specification from the executable code (see the rejection of claim 2 above).

With respect to claim 19, the claim is directed to a computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 17 (see the rejection of claim 17 above).

With respect to claim 20, the claim is directed to a method of facilitating static checking of executable code, the elements of which are analogous to those of claim 1 (see the rejection of claim 1 above).

With respect to claim 21, the claim is directed to a computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 20 (see the rejection of claim 20 above).

With respect to claim 22, the claim is directed to a data packet transmitted between two or more computer components that facilitates static checking of executable code, the elements of which are analogous to those of claim 1 (see the rejection of claim 1 above).

With respect to claim 23, the claim is directed to a data packet transmitted between two or more computer components that facilitates static checking of executable code, the elements of which are analogous to those of claim 1 (see the rejection of claim 1 above).

With respect to claim 24, the claim is directed to a computer readable medium storing computer executable components of an executable code check system, the elements of which are analogous to those of claim 1 (see the rejection of claim 1 above).

With respect to claim 25, the claim is directed to an executable code check system, the elements of which are analogous to those of claim 1 (see the rejection of claim 1 above).

9. Claims 9 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over DeLine in view of Rickel, as applied to claim 1 above, and further in view of U.S. Pub. No. 2004/0230958 to Alaluf ("Alaluf").

With respect to claim 9, the rejection of claim 1 is incorporated. DeLine in view of Rickel does not expressly disclose the object file being based, at least in part, upon a language that compiles to Common Language Runtime.

However, in an analogous art, Alaluf discloses performing static analysis on code that is based on a language that compiles to a Common Language Runtime (see, for example, paragraph [0038], lines 1-2, paragraph [0003], lines 1-3, and paragraph [0004], lines 8-9). Such code is platform and CPU independent (see, for example, paragraph [0003], lines 1-4, and paragraph [0004], lines 13-17).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of DeLine and Rickel to operate on an object file that is based, at least in part, upon a language that compiles to a Common Language Runtime, as Alaluf suggests. For example, one of ordinary skill in the art would have been motivated to modify the system of DeLine and Rickel such that it is platform and CPU independent.

With respect to claim 10, the rejection of claim 1 is incorporated. DeLine in view of Rickel does not expressly disclose the object file being based, at least in part, upon at least one of C#, Visual Basic.net and Managed C++.

However, in an analogous art, Alaluf discloses performing static analysis on code that is based on C#, Visual Basic.net or Managed C++, among others (see, for example, paragraph [0038], lines 1-2, and paragraph [0003], lines 8-11, and paragraph [0004], lines 8-9). Such code is platform and CPU independent (see, for example, paragraph [0003], lines 1-4, and paragraph [0004], lines 13-17).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of DeLine and Rickel to operate on an object file that is based, at least in part, upon at least one of C#, Visual Basic.net and Managed C++, as Alaluf suggests. For example, one of ordinary skill in the art would have been motivated to modify the system of DeLine and Rickel such that it is platform and CPU independent.

*Conclusion*

10. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure (see the attached Notice of References Cited).

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

My

Michael J. Yigdall  
Examiner  
Art Unit 2192

mjy



YIGDALL  
SUPERVISORY PATENT EXAMINER